10 คำถามเกี่ยวกับดีเบส

ยืน ภู่วรวรรณ*

หากคุณใช้ดีเบสเป็นประจำหรือบ่อย ๆ ละก็ คำถามต่อไปนี้คงเคยเกิดขึ้นในใจคุณ

ผู้เขียนได้รับคำถามจากทั้งทางจดหมาย โทรศัพท์ และเป็นการส่วนตัวอยู่บ่อยๆ เกี่ยวกับการใช้ดีเบส จึงขอรวบรวมนำมาอธิบายเพื่อให้ผู้สนใจทั่วไปได้เข้าใจและนำไปใช้ประโยชน์กันบ้าง สำหรับฉบับนี้ผู้เขียนจะนำมาทั้งหมด 10 คำถามที่น่าสนใจ

1. อยากทราบรายละเอียดวิธีการใช้และเทคนิคที่เกี่ยวกับ MEMO ฟิลด์

MEMO ฟิลด์ เป็นทางออกของดีเบสที่ดีมากทางหนึ่ง เพราะทำให้สร้างฟิลด์ที่เป็นแบบ variable length ได้ โดยความจริงแล้วดีเบสจะกันเป็นบล็อกๆ สำหรับเก็บข้อความในฟิลด์นี้ โดยเก็บแยกอีกแฟ้มหนึ่งโดยใช้ .dbt

เมื่อสร้างฟิลด์ชนิด MEMO ขึ้นมา จะเสียเนื้อที่ใน dbf ไปเรคอร์ดละ 10 ไบท์ และถ้าไม่มีการใช้ฟิลด์นี้คือไม่มีข้อมูลก็จะเสมือนมี 0 ไบท์ใน dbt แต่ถ้ามีการใช้ข้อมูลก็จะกำหนดขึ้นเป็นบล็อก โดยค่ากำหนดของดีเบสทรีใช้ 512 ไบท์ต่อบล็อก โดยรวมกันแล้วในหนึ่งฟิลด์นี้ต้องไม่เกิน 5,000 ไบท์ แต่สำหรับดีเบสทรีพลัส สามารถกำหนดขนาดความกว้างได้ด้วยคำสั่ง SET MEMOWIDTH TO xx โดยตัวเลข xx บอกขนาดของความกว้างในการแสดงผล

การใช้เมโมฟิลด์นี้ จึงเป็นการเก็บข้อความ เช่น Abstract ของหนังสือ บัตรบันทึกคนไข้หรือหมายเหตุอื่น ๆ

สิ่งที่น่าสังเกตคือ เราจะดูข้อมูลในเมโมฟิลด์ได้ 2 วิธีคือ

- ใช้ GOTO ไปเรคอร์ดที่ต้องการแล้วเรียก EDIT จากนั้นเลื่อนเคอร์เซอร์มาที่ฟิลด์เมโม กด Ctrl-PgDn เพื่อขอเข้าไปแก้ไข
- ใช้คำสั่ง LIST หรือ DISPLAY MEMO ซึ่งก็จะได้ตามที่ต้องการเช่นกัน

ในการเขียนด้วยโปรแกรมเราอยากมีคำสั่ง REPLACE ระหว่างตัวแปรกับเมโมฟิลด์ ปรากฏว่าดีเบสทรีทำไม่ได้ แต่สำหรับ CLIPPER ดีเบสคอมพายเลอร์ยอมให้เราใช้คำสั่ง REPLACE MEMOFIELD WITH ตัวแปร หรือ STORE MEMOFIELD TO ตัวแปร ได้ จึงทำให้เราใช้งานได้ดีมากและตรงกับความต้องการของผู้เขียนโปรแกรมมาก

2. อยากทราบประโยชน์ HISTORY ที่มีมาในดีเบสทรีพลัส

การใช้ HISTORY นับเป็นสิ่งที่ดีมากของดีเบสทรีพลัส และเป็นสิ่งที่เพิ่มเติมเข้ามาให้ผู้ใช้ได้ประโยชน์อย่างมาก การใช้ HISTORY มีจุดมุ่งหมาย 2 ประการคือ

ประการแรกคือมีไว้เก็บคำสั่งที่เคยใช้มาแล้วมาใช้อีก ทุกครั้งที่อยู่ที่ . เมื่อคีย์คำสั่งใดแล้วตามด้วยรีเทอร์น ดีเบสจะตีความหมายคำสั่งนั้นแล้วนำคำสั่งนั้นเข้าเก็บไว้ในตารางประวัติ ซึ่งเก็บคำสั่งย้อนหลังไว้ 20 คำสั่งที่คีย์ล่าสุด ดังนั้นผู้ใช้จึงสามารถเรียกมาแก้ไขคำสั่งได้ง่าย ผู้ใช้ลองดูโดยการกดคีย์ ↑ ขณะอยู่ที่ . จะเห็นว่ามีคำสั่งที่เคยคีย์มาแล้วปรากฏออกมา และเราสามารถใช้คีย์บอร์ดทำการแก้ไขได้

การเก็บ HISTORY ดีเบสเก็บไว้ตามปกติ 20 คำสั่ง และสามารถใช้คีย์บอร์ด ↑ ↓ เรียกคำสั่งที่เก็บไว้ใน HISTORY มาแก้ไขได้ การแก้ไขคำสั่งจะใช้คีย์บอร์ดต่อไปนี้ ได้คือ

→	เลื่อนเคอร์เซอร์ไปทางขวาทีละตัว
←	เลื่อนเคอร์เซอร์ไปทางซ้ายทีละตัว
HOME	เลื่อนไปต้นคำของคำสั่ง
END	เลื่อนไปท้ายคำสั่ง
BACKSPACE	ลบตัวอักษรทางซ้ายเคอร์เซอร์
DEL	ลบตัวอักษรตำแหน่งเคอร์เซอร์

*ห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์
ภาควิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเกษตรศาสตร์

INS เปลี่ยนมาเป็นโหมด INS เพื่อแทรกสิ่งที่จะคีย์

การแก้ไขคำสั่งจาก HISTORY นี้ มีประโยชน์มากลองใช้ดูแล้วจะเห็นคุณค่า

ประการที่สอง คือผู้ใช้สามารถ SET-HISTORY TO xx เมื่อ xx เป็นตัวเลขที่บอกถึงคำสั่งที่จะเก็บใน HISTORY ได้ ซึ่งจะเซ็ตให้มากกว่า 20 ก็ได้ และมีคำสั่ง LIST HISTORY หรือ DISPLAY HISTORY เพื่อขอดู ดีเบสจึงให้ HISTORY เสมือนเป็นการดีบักโปรแกรมได้ โดยการ RUN โปรแกรม ตามปกติจะไม่เก็บใน HISTORY จะเก็บเฉพาะที่คีย์ที่ . เท่านั้น ดีเบสให้ใช้ SET DOHISTORY ON เพื่อจะเก็บคำสั่งที่ RUN ใน .PRG ใน HISTORY เราจึงสามารถใช้ HISTORY ในการดูลำดับการทำงานของโปรแกรมเพื่อการตรวจสอบได้ ว่าดีเบสทำงานในโปรแกรมมีลำดับอย่างไร

3. อยากทราบวิธีการโอนย้ายแฟ้มข้อมูลจากดีเบสไปโลตัสเพื่อเขียนกราฟ

วิธีนี้ทำได้ง่ายมากในดีเบสทรีพลัส สมมติว่า เรามีข้อมูลอยู่ 2 ฟิลด์ ฟิลด์แรกเป็นเดือน ฟิลด์ที่ 2 เป็นยอดขาย และเก็บข้อมูลไว้ในแฟ้มชื่อ SALE เราทำได้ดังนี้

```
.USE SALE
.COPY TO SALE WKS
```

หากลอง DIR ดูจะเห็นแฟ้ม SALE.WKS เกิดขึ้น ซึ่งเป็นแฟ้มที่สามารถอ่านด้วยโลตัสได้โดยตรง โดยข้อมูลในเซล A1 เป็นชื่อฟิลด์ และคอลัมน์ A จะเป็นฟิลด์แรกคอลัมน์ B จะเป็นฟิลด์ที่ 2

ทำนองเดียวกันถ้าต้องการอ่านข้อมูล WKS มาไว้ในดีเบสก็ทำได้ด้วยคำสั่ง

```
.APPEND FROM SALE WKS
```

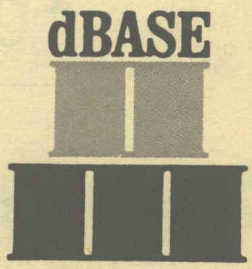
4. อยากทราบประโยชน์ของคำสั่ง MODIFY SCREEN

ผู้เขียนโปรแกรมหลายท่านอาจอึดอัดใจในการออกแบบสกรีนสำหรับการทำ DATA ENTRY เพราะอาจต้องออกแบบในกระดาษ กำหนดแถวและคอลัมน์ให้ถูกต้องแล้วนำมาเขียนโปรแกรม @ ROW, col SAY ซึ่งถ้ามีการขยับหรือแก้ไขจะยุ่งยากพอควร ในกรณีนี้ดีเบสมี program generator ช่วยเขียนโปรแกรมให้เราแล้ว และเป็นตัวสร้างสกรีนที่ดีมากตัวหนึ่ง ผู้เขียนชอบใช้มาก เพราะเมื่อสร้างสกรีนเสร็จแล้ว เราสามารถผนวกโปรแกรมที่ดีเบสเขียนเข้ากับส่วนของโปรแกรมของเราโดยใช้ไซด์คิกอ่านบล็อกเข้ามารวมได้เลย

ขั้นตอนของการสร้างสกรีนทำได้โดย

1. เรียก MODIFY SCREEN (ชื่อแฟ้ม)
2. เข้ามาสู่เมนูของ screen editing ถ้าต้องการโหลดแฟ้มใดเพื่อทำ data entry ให้บอกและกำหนดฟิลด์ที่ต้องการ
3. เข้าสู่ blackboard ด้วย F10 หลังจากนั้นสร้างสกรีนด้วยฟูลสกรีนอีดิเตอร์ สามารถ drag field ไปที่ใดก็ได้ สามารถตีกรอบ สร้างรูปแบบเส้นได้ตามต้องการ.
4. เมื่อ Save จะได้แฟ้ม .FMT ลอง TYPE แฟ้มนี้ดูจะเห็นเป็นโปรแกรมดีเบสที่เราสามารถนำไปผนวกกับโปรแกรมที่เราเขียนได้

ลองดูสิครับแล้วจะเห็นว่าสร้างสกรีนของดีเบสนี้ง่ายมาก และที่สำคัญคือ เราใส่รหัสภาษาไทยในสกรีนนี้ก็ได้ครับ เพราะดีเบสรับ 8 บิท



5. ในการค้นหาข้อมูลมีคำสั่ง FIND SEEK LOCATE และจะตรวจสอบสถานะการพบข้อมูลได้อย่างไร

ในแฟ้มข้อมูลชนิด INDEX เราจะค้นหาโดยใช้ FIND หรือ SEEK โดย FIND นั้น จะต้องหาชื่อโดยตรง เช่น FIND DANG หมายถึงหาข้อมูล DANG ดังนั้นถ้าข้อมูล DANG อยู่ในตัวแปร เช่น A = "DANG" จะต้องใช้ FIND &A แทน ส่วน SEEK นั้น ใช้กับตัวแปรได้เลยคือ SEEK A

หากหาไม่พบจะได้ EOF( ) เป็น .T. ดังนั้นเราตรวจสอบด้วย

```
IF EOF( )
  ? "CAN NOT FOUND"
ELSE
  ? "FOUND!"
ENDIF
```

ส่วน LOCATE เป็นการหาแบบลำดับ เราใช้ GO TOP แล้ว LOCATE FOR (ชื่อฟิลด์) = (ตัวแปร) การหาไม่พบก็จะได้ EOF( ) เป็น .T. เช่นกัน

ในดีเบสทรีพลัส เราสามารถใช้ FOUND( ) แทนได้ เช่น

```
IF FOUND ( )
  ? "FOUND! ."
ELSE
  ? "CAN NOT FOUND"
ENDIF
```

6. ผมมีข้อมูลที่เขียนไว้แล้วในโปรแกรมสำเร็จรูปอื่น เช่น มัลติแพลน พีเอฟเอส จะนำมาใช้ในดีเบสได้อย่างไร

เช่นเดียวกับคำถามข้อ 3 ถ้าต้องการโอนย้ายโดยตรงเราใช้ COPY และ APPEND ได้โดยตรง เช่น

.COPY TO (ชื่อไฟล์) SYLK

จะได้แฟ้ม ที่มัลติแพลนอ่านได้ (อ่านแบบ SYMBOLIC LINK) และก็สามารถใช้ APPEND ได้เช่นกัน

สำหรับพีเอฟเอสนั้น เรามีคำสั่งใหม่เพิ่มมาคือ

.EXPORT TO (ชื่อไฟล์)
TYPE PFS

ซึ่งจะสร้างแฟ้มเป็น PFS ไฟล์ทันที และถ้าต้องการอ่านแฟ้ม PFS ก็ใช้

.IMPORT FROM (ชื่อไฟล์)
TYPE PFS

7. การสร้างช่องหน้าต่างในดีเบสมีหลักการอย่างไร

ดีเบสทรีพลัส มีคำสั่งตีกรอบอยู่แล้วโดยใช้

@ row1, col1 TO row2, col2 (double)

หากไม่เติมคำว่า double จะได้กรอบสี่เหลี่ยมเส้นเดี่ยว แต่ถ้าเติม double จะได้กรอบสี่เหลี่ยมเส้นคู่

@ row1, col1 TO row2, col2 clear

เป็นคำสั่งการลบช่องหน้าต่าง ลองดูคำสั่งต่อไปนี้

```
CLEAR
@ 3, 5 TO 10, 40
@ 7, 20 SAY "WINDOW1"
@ 22, 0
WAIT
@ 8, 25 TO 15, 70 CLEAR
@ 8, 25 TO 15, 70 DOUBLE
@ 13, 50 SAY "WINDOW2"
@ 22, 0
WAIT
```

8. อยากทราบโครงสร้างแฟ้มของ .dbt เพื่อจะไปเชื่อมกับโปรแกรมอื่นได้

ปกติในแฟ้มข้อมูล .dbf ถ้าใช้ดีบักดูจะพบว่าไบท์แรกของแฟ้มจะบอกแฟ้มว่าเป็น ดีเบสทรี หรือทู โดยที่ถ้ามี

02 หมายถึง ดีเบสทู
03 หมายถึง ดีเบสทรี
83 หมายถึง ดีเบสทรี ที่มี dbt

ส่วนแฟ้ม .dbt จะเก็บข้อมูลเป็นบล็อก โดยทั่วไปจะเก็บบล็อกละ 512 ไบท์ เรียงกันเป็นลำดับ

ในบล็อกแรกจะเป็น header และเราเรียกว่าเป็นบล็อก 0 และบล็อกต่อมาเป็นบล็อก 1, 2, 3.....ดีเบสทรีพลัส จะใช้ 4 ไบท์แรกของ header เป็นตัวบอกจำนวนบล็อกที่มีในแฟ้มข้อมูลนี้ โดยเก็บเป็นตัวเลขฐานสิบหก โดยไบท์แรกเป็นไบท์ที่มีนัยสำคัญน้อยที่สุด เช่น ถ้าแฟ้ม .dbt มี 200 บล็อก ก็จะเก็บ C8 00 00 00 และถ้ามี 4321 บล็อก ก็จะเก็บ E1 10 00 00

ใส่ส่วนของแฟ้ม dbf ในส่วน MEMO FIELD จะเก็บเฉพาะหมายเลขบล็อกของข้อมูลไว้เท่านั้น ว่าเรคอร์ดนี้มีฟิลด์เมโมอยู่ที่บล็อกที่เท่าใดใน dbt ข้อมูลในแต่ละบล็อกจะปิดท้ายด้วย 1A บอกว่าหมดฟิลด์นี้แล้ว และหากเมโมฟิลด์ใดยาวเกินกว่าบล็อก เช่น 520 ไบท์ มันจะใช้ 1 บล็อก 512 ไบท์ และบล็อกต่อไปอีก 8 ไบท์ ตามด้วย 1A อีก 2 ไบท์ หลังจากนั้นในบล็อกนี้จะไม่ใช้เลย

หากใช้ดีบักดูข้อมูลในเมโมฟิลด์ของ dbf จะเห็น ASCII ของตัวเลขบอกหมายเลขบล็อกเอาไว้ ในการแก้ไขข้อมูลใน dbt ดีเบสจะถือว่าจะไปสร้างบล็อกใหม่ต่อท้ายเพื่อให้ขยายเพิ่มบล็อกได้ และจะละทิ้งข้อมูลเดิมไว้ โดยเปลี่ยนเฉพาะตัวชี้ที่อยู่ใน dbf มาเท่านั้น ดังนั้นจะเห็นว่าแฟ้ม dbt จะโตเร็วมาก

เราสามารถจัดแฟ้ม dbt ใหม่โดยใช้คำสั่ง COPY TO ในการ COPY .dbf ซึ่งจะมีการจัดแฟ้ม dbt ใหม่ด้วย

9. จะทำให้เกิดเสียงปี๊ปเตือนการทำงานบางขณะได้อย่างไร

ดีเบส ให้เราใช้ CHR(7) ได้อยู่แล้ว ลองใช้

```
.?CHR(7)+'Be more careful'
```

10. ผมใช้งานดีเบสมามาก รู้สึกเบื่อ prompt อยากเปลี่ยนเป็นตัวอื่นทำได้อย่างไร

ดีเบสทรีพลัสยอมให้สร้างแฟ้ม CONFIG.DB โดยเมื่อดีเบสถูกเรียกเข้ามามันจะดูว่ามีแฟ้ม CONFIG.DB หรือไม่ ถ้ามีมันจะนำมาทำงาน ผู้ใช้สามารถเซ็ทอะไรลงไปใน CONFIG.DB ได้ เช่น ถ้าต้องการ prompt เป็นตัวอื่นก็ใส่คำสั่ง

prompt = "$"

ผลจะได้ . เป็น $ แทน

ใส่ส่วนของ CONFIG.DB นี้เราจะใส่คำสั่ง SET ต่าง ๆ ได้อีกมากลองพยายามดูจะเห็นว่าดีเบสช่วยเราให้เกิดความคล่องตัวในการทำงานอยู่แล้ว